School Computing, Faculty of Engineering
Universiti Teknologi Malaysia

SCSJ1023 Programming Technique II
Semester 2, 2018/2019

Individual Report for the Group Project

Notes: *This report is to be submitted individually.*

Student's Name	Abdullah AlSolaiman
Group Project Title	**The Pharmacy**

A. Implementation of Classes

Describe the class (or classes) you implemented for the project. Give the estimate percentage of how much you did on the class. For example, if you did all the code for the class, then you write 100%. Also, give the location of the code you did.

Class Name	Percentage of contribution	Location		Remarks
		File	Line Numbers	
Date	**100%**	**Afile.cpp**	**18-36**	Default Argument Constructor because it is a good practice to always initialize the members.
Medicine	50%	Afile.cpp	37-55	**Const for the getters is good practice**
Pharmacy	80%	Afile.cpp	93-137	**Iequals used for incase sensitive compare**
Patient	50%	Afile.cpp	191-214	**Asigning history to use it in outputfile**
Person	0%	Afile.cpp	-	-
Doctor	20%	Afile.cpp	250-264	**Polymorphisim of the function print()**

B. Implementation of OOP Concepts

Describe the concept (or concepts) you implemented for the project. Give the estimate percentage of how much you did on the concept. Explain why the concept is needed in your project, and explain the general idea of how you implemented it.

*The concepts to be described here include **Association, Inheritance and Polymorphism**. Note that, each member of a group is not necessarily to implement all the concepts.*

OOP Concept	Percentage of contribution	Location		Why is this concept needed?	General idea of the implementation
		File	Line Num.		
Polymorphisim	**80%**	Afile.cpp	**200 & 251**	**to have the ability (in programming) to present the same interface for differing underlying forms (data types).**	**The function print() when called from different classes will output different info based on the class it called from.**
Inheritance	50%	Afile.cpp	216 & 159	**To have specialized classes that have attributes and methods of their own, and in the same time inherit the attributes and methods from the parent class**	**Class Patient and class Doctor both inherit the name and the id**
Aggregation	20%	Afile.cpp	95	**The life span of the object is not dependening on the class containing it.**	**The class dr having a list of patients, if for some reason dr has to be deleted, that will not delete the list of patients, it will only UNLINK them from doctor.**

Composition	50%	Afile.cpp	40 & 166	**The life span of the object is dependening on the class containing it.**	**If Medicine gets deleted, then ofcourse the object dateOfExpiry will be of no use, so it will be deleted automatically when the Medicine is deleted**
Encapsulation	100%	Afile.cpp	-	**it provides your code - security, flexibility and its easy maintainability**	**All Classes are gathering different attributes and methods like a collection**
Data hiding	100%	Afile.cpp	-	**useful in hiding the data(instance variables) of a class from an illegal direct access**	**All private members of classes cannot be accessed directly by the object so that will always keep them secured from illegal modification.**

C. Other Implementations (Optional)

This part is only to be filled in should you have other things you did for your project but have not been mentioned in Part A and B.

Things / Code Done	Percentage of contribution	Remarks
Class Art	100%	Used to encapsulate methods to print a string into an ASCII art on the screen, welcoming messages for example. See #inclue "Art.h"
Function iequals	100%	function to compare two string not-case senstive (like asPiRin)
Curses library	100%	Useful library to create user interface menu that is interactive with the user.